YOORshop SAS
RCS 817 466 147 — 254 Rue Vendôme, 69003 Lyon, France — yoorhosting.com
[informations de société à compléter]

Réf. rapport : YH-DEMO-2026-02
Date d'analyse : 30 octobre 2026

● DIAGNOSTIC & SOLUTIONS CLÉS

RAPPORT D'INGÉNIERIE APPLICATIVE & BASE DE DONNÉES • LIVRABLE N°2

Optimisation des performances d'un site

Analyse des pics CPU à 70 %, contentions de cache de menu et réécriture SQL d'un balayage de 787 504 lignes.

CLIENT DESTINATAIRE

Société : SARL Moto Pièces Express (Fictif)
Boutique en ligne : https://boutique-moto.example (PrestaShop 8.2.3)
Serveur dédié : srv-demo.yoorhosting.com (Xeon 16 cœurs, 64 Go RAM)
Environnement : PHP-FPM pool motodemo_prod (48 workers) • MariaDB

CHARGE CPU OBSERVÉE
70 % à 1 270 %
45 pics enregistrés lors des imports

REQUÊTES LENTES PHP
540 / 976
Bloquées dans preg_grep du menu

ACCÉLÉRATION SQL CONSTATÉE
x 4 000
De 1,60 s à 0,0004 s par appel

TEMPS FINAL REQUÊTE NOUVEAUX PRODUITS
0,0004 s
Par comparaison directe indexée

DEMANDE DU CLIENT

« Constatant des ralentissements majeurs lors de l'injection régulière de tarifs et de stocks via des modules d'import automatisés, avec des saturations processeur récurrentes (pics jusqu'à 1 270 % de CPU cumulé sur 16 cœurs), le client a mandaté YOORshop pour **identifier l'origine précise des blocages** et prescrire un **plan d'action d'optimisation sans impact négatif sur le catalogue**. »

MÉCANISME COMMUN DES SURCHARGES IDENTIFIÉ

L'analyse croisée du journal des requêtes lentes PHP (*PHP slowlog*) et des horodatages de modification de la base (`date_upd`) a mis en évidence un goulot d'étranglement structurel : **chaque enregistrement d'un produit déclenche la purge simultanée de deux caches applicatifs lourds**, qui sont immédiatement recalculés en boucle lors des flux de synchronisation :

Corrélation temporelle établie : La modification de 22 produits en une heure coïncide à la seconde près avec l'explosion des requêtes lentes (20:44, 20:47, 20:49, 20:50, puis 20:52 à 20:56). Au total, 45 pics de charge ont été enregistrés, déclenchés par les requêtes POST d'administration exécutées depuis l'adresse IP opérateur `198.51.100.22` .

1 Problème N°1 : Régénération abusive du menu (Module `ps_mainmenu`)

POINT D'ANALYSE	CONSTAT TECHNIQUE & SOLUTION VALIDÉE
Accrochage excessif	Le module est inutilement greffé sur `actionObjectProductAddAfter` , `actionObjectProductUpdateAfter` et `actionObjectProductDeleteAfter` (table `psdemo_hook_module`). Ces hooks forcent l'appel à `clearMenuCache()` (ps_mainmenu.php:1007), détruisant l'intégralité du cache JSON.
Coût de reconstruction	**540 des 976 requêtes lentes** PHP sont bloquées dans `preg_grep` (ps_mainmenu.php:796). La reconstruction compare **9 788 catégories à 26 968 images**. Deux processus concurrents recalculent le même fichier en parallèle (conflit d'écriture à 21:02:01 et 21:02:05).
Remédiation immédiate	**Détacher `ps_mainmenu` des trois hooks produit** depuis le back-office PrestaShop (*Apparence* > *Positions*). Modification 100 % sans risque : le menu ne contient aucun produit individuel (configuration `MOD_BLOCKTOPMENU_ITEMS = CAT77,CAT71,LNK1,LNK3`). Les hooks catégorie et pages CMS restent actifs.

2 Problème N°2 : Requête non indexée du bloc « Nouveaux produits » (`ps_newproducts`)

Le module `ps_newproducts` exécute après chaque mise à jour la méthode `Product::getNewProducts()`. L'écriture d'origine applique un calcul direct sur la colonne indexée, rendant l'index inutilisable par MariaDB :

Requête d'origine (Non sargable) **1,60 seconde**

```
WHERE DATEDIFF(date_add, NOW()) < 20
```

Conséquence MariaDB : L'index `date_add` est désactivé. *Full Table Scan* de **787 504 lignes** dans `psdemo_category_product`, création d'une table temporaire sur disque et tri lourd.

✓ Requête optimisée (Surcharge override) **0,0004 seconde**

```
WHERE date_add > '2026-10-07'
```

Conséquence MariaDB : Utilisation immédiate de l'index B-Tree. Balayage restreint aux 8 enregistrements requis. **Gain de rapidité mesuré : facteur 4 000**.

3 Dimensionnement des capacités du pool PHP-FPM

Pour absorber les flux simultanés sans refoulement HTTP 503, le pool `motodemo_prod` a été dimensionné sur mesure par nos équipes :

PROCESSUS MAX (CHILDREN)
48 workers
Standard mutualisé : 15

MÉMOIRE ALLOUÉE
1 024 Mo / worker
Absorption des imports catalogue

CHARGE CRON MOYENNE
57 % d'un cœur
Régime stabilisé à la minute

RESTAURATION DES CACHES
Immédiate
Fin des conflits de processus

4 Synthèse des actions & Gains attendus

#	ACTION PRÉCONISÉE	MODE OPÉRATOIRE	IMPACT MESURÉ
1	Détachement du module `ps_mainmenu`	Back-office (Apparence > Positions)	**-540 requêtes lentes** Suppression de la contention `preg_grep`.
2	Déploiement surcharge SQL `getNewProducts`	Fichier dans `override/classes/Product.php`	**Temps divisé par 4 000** Fin du scan de 787 504 lignes MariaDB.
3	Purge des sondes de surveillance	Nettoyage de `/root/cpuwatch/`	**Terminé** Sondes arrêtées le 28/10 à 12:44.

Validation technique d'expertise :
Pôle Ingénierie & Performance Web — YOORshop SAS

✓ Plan d'Optimisation Validé